

# CANoe/CANalyzer Workshop for CAN and LIN

<b>Duration:</b>	3 Days
<b>Target group:</b>	CAN and LIN users (controller development, motor vehicle electrical, test planning and execution)
<b>Prerequisites:</b>	none

## Day 1: CAN and LIN Fundamentals

### 1 Introduction to serial bus systems in motor vehicles (1.0 h)

Goal: Build an understanding of serial communications in the motor vehicle

Contents: Motivation for the use of serial communication systems, properties of serial communication systems

### 2 CAN Protocol (2.0 h)

Goal: Understanding of the basic principles of the CAN protocol

Contents: Bus levels, signal encoding, signal delay, synchronization, resynchronization, bit time interval, bit stuffing, bus access methods, message transmission, message formats, error detection and error handling, failure detection, future prospects, exercises

### 3 CAN Data Interpretation with CANdb++ (1.0 h)

Goal: Interpretation of CAN data using the CAN database

Contents: CANdb++ Editor: Network nodes, messages, signals, conversion formulas, exercises

### 4 Fundamentals of the LIN Protocol (2.0 h)

Goals: Build up an understanding of LIN protocol basics

Contents: LIN network structure, principle of communication, framing (frame types, frame slots), scheduling, task behavior model, network management, status management, synchronization, data save, milestones of LIN development, LIN Specification

### 5 Fundamentals of the LIN Network Description and Configuration (1,0 h)

Goals: Build up an understanding of LIN network description and configuration

Contents: System defining process, functions and services, LIN node capability language specification.

## Day 2: Analysis & Stimulation with CANalyzer and CANoe

### 6 LIN Data Interpretation with LIN Network Viewer (1.0 h)

Objective: Interpretation of LIN network description (LIN Description File LDF) with help of the LIN Network Viewer.

Contents: LIN Network Viewer: Network nodes, messages, signals, conversion formulas, schedule tables, special LIN attributes

## 7 Measurement, Evaluation and Data Logging with CANoe/CANalyzer (3.0 h)

Goal: Use of CANoe and CANalyzer as measurement tools

Contents: Configuration of windows and function blocks, data tracing, Statistics window, signal analysis in Data and Graphic windows, present logging functionality, use of specific trigger conditions, exercises

## 8 Stimulation and Emulation (2.0 h)

Goal: Active intervention into existing bus traffic by means of Send blocks

Contents: Generator blocks and Replay block, exercises

## 9 Offline Analysis (1.0 h)

Goal: Explain Offline mode

Contents: Offline analysis of logged data in Offline mode, exercises

## Day 3: CAPL Programming and Modelling with CANoe

### 10 Introduction to CAPL (1.5 h)

Goal: Provide overview of the CAPL programming language (CAN Access Programming Language) for modelling network nodes

Contents: Event procedure concepts of CAPL, signal analysis in CAPL, sending messages in CAPL, exercises

### 11 Panels in CANoe (1.0 h)

Goal: Introduce the Panel Editor that is used to create and modify control and display panels

Contents: Environment variables, display and control elements, bitmap control, pointer control, overlapping of controls, copying and pasting controls, exercises

### 12 LIN Cluster Simulation (3.0 h)

Goal: Simulate LIN cluster nodes with help of CAPL and panels

Contents: Simulation of slave tasks for LIN 2.0 clusters, simulation of master task, scheduling of unconditional, diagnostic, sporadic and event triggered frames, network management, exercises

### 13 LIN Slave Conformance Tester in CANoe (1.0 h)

Goal: Introduce the Panel Editor that is used to create and modify control and display panels

Contents: Environment variables, display and control elements, bitmap control, pointer control, overlapping of controls, copying and pasting controls, exercises

### 14 Tips and Tricks in Working with CANoe Models (0.5 h)

Goal: Discussion of special questions and problems in using CANoe

Contents: Working with directories, working with multiple databases, CANoe internal program structure, performance optimization, working with configurations